REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-13 are pending in the application. The Examiner additionally stated that claims 1-13 are rejected. By this amendment, claims 1-13 have been amended and claims 14-20 have been added. Hence, claims 1-20 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

Information Disclosure Statement

In the Office Action, the Examiner noted that Applicant filed an IDS on March 9, 2004, and did not provide hard copies for several of the disclosed references.. Because of this, the Examiner did not consider those references. The Examiner provided copies of the March 9, 2004 IDS in the instant Office Action and indicated those references that were not considered by strikethrough markings thereon. In response, Applicant herewith submits an Information Disclosure Statement (including IDS Transmittal, Information Disclosure Statement By Applicant, copies of IDS citations, and corresponding fee per 37 CFR 1.17(p)) providing copies of the noted references.

In the Specification

The Examiner objected to the specification, noting that the title of the invention is not descriptive, and requiring a new title that is clearly indicative of the invention to which the claims are directed. Applicant has amended the title with a new title that clearly indicates the invention to which the claims are directed. Accordingly, it is requested that the objection to the specification be withdrawn.

In the Claims

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claims 1-13 under 35 U.S.C. §103(a) as being unpatentable over Parady, U.S. Patent No. 5,933,627 (hereinafter, "Parady"), in view of McFarling et al., U.S. Patent No. 5,758,142 (hereinafter, "McFarling"). Applicant respectfully traverses the Examiner's rejections.

Prior to providing a claim-by-claim analysis, a brief summary of the teachings of Parady and McFarling are provided below, vis-à-vis the invention disclosed by Applicant in the instant application. This information is provided to aid the Examiner during reconsideration of the claims.

Parady discloses a method and apparatus for switching between threads of a program *in response to a long-latency event.* In one embodiment, the long-latency events are load or store operations which trigger a thread switch *if there is a miss* in the level 2 cache. In addition to providing separate groups of registers for multiple threads, a group of program address registers pointing to different threads are provided. A switching mechanism switches between the program address registers *in response to* the long-latency events. (cf. Abstract) Parady also defines the process whereby a multithreading processor interleaves threads in such a manner as described above (i.e., in response to a long-latency event) as "coarse-grain multithreading," (cf. col. 2, lines 8-10) and furthermore teaches the concept of "a switching mechanism [that] switches between the program address registers *in response to* the long-latency events. Parady discloses the switching mechanism as "[t]hread switching logic 112 provided to give a hardware thread switching capability. The indication that a thread switch is required is provided on a line 114 providing an *L2-miss indication from cache control/system interface 22.*" He further teaches that "*[u]pon such an indicatio*n, a switch to the next thread will be performed." (cf. col. 3, lines 57-62) Furthermore, in a background discussion of his invention, Parady cites an IBM article that distinguishes between processors to which his invention is directed (i.e., coarse-grain multithreaded processors) and fine-grain multithreaded processors, that is, those processors which interleave threads on a cycle-by-cycle basis. (cf. col. 2, lines 6-10) Parady's invention and disclosure is directed towards problems associated with coarse-grain processors: those processor that switch threads *in response to* long latency events.

McFarling discloses a predictor which chooses between two or more predictors. The predictor includes a first component predictor which operates according to a first algorithm to produce a prediction of an action and a second component predictor which operates to according to a second algorithm to produce a prediction of said action. The

predictor also includes means, coupled to each of said first and second predictors, for choosing between predictions provided from said predictors to provide a prediction of the action from the predictor. The predictor can be used to predict the outcomes of branches, cache hits, prefetched instruction sequences, and so forth. (cf. Abstract) In a description of the invention, McFarling discloses a mux 24 that provides either a branch prediction from predictor 12 or predictor 16. (cf. col. 6, lines 12-14) This mux 24 is shown in embodiments described with reference to figures 2 and 5, and is also shown as mux 54 in figure 7, which employs the output 54b of 1st cache hit predictor 52 and 2nd cache hit predictor 56 based upon input from signal cache_hit provided to choosing logic 62. It is clear from the presence of a single mux 24, 54, that McFarling's teachings address prediction in a single-threaded processor. Nowhere therein does McFarling suggest, or provide any other motivation to one skilled in the art, that such teachings may be applied in a multithreaded context.

In contrast to the teachings of Parady and McFarling, Applicant's invention is directed towards a processor having multiple hardware streams supporting multiple data threads, and a data cache. In this processor, Applicant discloses a system for fetching instructions from a selected one of the multiple hardware streams to a pipeline. The system includes multiple hit/miss predictors and a fetch algorithm. The multiple hit/miss predictors are each associated with a corresponding one of the multiple hardware streams, and each forecasts whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache. The fetch algorithm is coupled to the multiple hit/miss predictors. The fetch algorithm selects, *on a cycle-by-cycle basis*, the selected one of the multiple hardware streams from which to fetch the instructions.

By Parady's own admission, his invention is directed towards coarse-grain multithreading applications. In these applications and corresponding multithreaded processors, thread switches occur *in response to long-latency events.* Such a configuration is evidenced by signal line 114 in Parady that provides an L2-miss indication from cache control/system interface 22. By stating a specific multithreading application (i.e., coarse-grain multithreading) to which his invention is directed, in contrast to the multithreading application to which Applicant's invention is directed (i.e.,

fine-grain multithreading, that switches threads on a cycle-to-cycle basis), Parady thus teaches away from the present invention. It therefore does not follow that the teachings of Parady and McFarling can be combined in a manner relative to Applicant's invention because Parady's invention is directed towards *responding to the occurrence of long-latency events* and McFarling's invention is directed towards *preclusion of the occurrence of long-latency events*.

In view of the above summarizations, a claim-by-claim analysis will now be presented.

Amended claim 1 is provided below for ease of reference.

1.: In a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from a selected one of the multiple hardware streams to a pipeline, the system comprising:

multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache; and

a fetch algorithm, coupled to said multiple hit/miss predictors, configured to select, on a cycle-by-cycle basis, the selected one of the multiple hardware streams from which to fetch the instructions.

Claim 1 recites, in combination, within a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from a selected one of the multiple hardware streams to a pipeline. The system has multiple hit/miss predictors that are each associated with a corresponding one of the multiple hardware streams. In addition, each of the multiple hit/miss predictors is configured to forecast whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache. The system also has a fetch algorithm that is coupled to the multiple hit/miss predictors. The fetch algorithm selects, *on a cycle-by-cycle basis,* the selected one of the multiple hardware streams from which to fetch the instructions.

In the rejection of claim 1, the Examiner notes that Parady has taught in a processor having multiple hardware streams supporting multiple date threads, and a date cache, a system for fetching individual ones of the multiple streams to a pipeline, comprising: a) a fetch algorithm for selecting from which stream to fetch instructions. The Examiner noted that Parady has not taught a hit/miss predictor for forecasting whether instructions will hit or miss the data cache wherein the prediction by the hit/miss predictor is used by the fetch algorithm in determining from which stream to fetch. The Examiner stated that McFarling, however, has taught a hit/miss predictor that is used to predict, for load, instructions, whether a cache hit or miss will occur and, if a cache miss is predicted to occur, then instruction independent of the load are schedule ahead of the load-dependent instructions. The Examiner opined that a person of ordinary skill in the art would have recognized that this prediction scheme would be useful in a multiple-thread environment because a thread is a sequence of instructions that is independent from other threads, as is know in the art. The Examiner further noted that by implementing such a prediction scheme into the system of Parady, thread switches can occur sooner, thereby maximizing efficiency through execution of load-independent instructions. The Examiner concluded that it would consequently have been obvious to one of ordinary skill in the art at the time of the invention to modify the thread-switching system of Parady to include a hit/miss predictor as taught by McFarling, in order to switch thread before the load reaches the execution stage, thereby preventing a drop-off in throughput.

Applicant respectfully disagrees with the Examiner's arguments summarized above for because claim 1 recites a fetch algorithm that selects, *on a cycle-by-cycle basis*, the selected one of the multiple hardware streams from which to fetch instructions. Parady discloses that thread switches occur *in response to long-latency events.* Applicant has thoroughly searched the teachings of Parady to find any motivation, suggestion, or even a hint that the teachings therein could be applied in a fine-grain multithreading context. The only mention of such context that Applicant finds is that Parady mentions fine-grain multithreading in a background discussion to distinguish this area of the art from the area to which his invention is directed (i.e., coarse-grain multithreading). In addition, Applicant respectfully disagrees with the Examiner's point that by implementing

McFarling's prediction scheme into the system of Parady, thread switches can occur sooner, thereby maximizing efficiency through execution of load-independent instructions. This does not follow, because, according to Parady's own teachings, McFarlings prediction would only be noted after line 114 is asserted indicated a cache miss. Parady's inventive concept is predicated by the existence of long-latency events. He teaches nothing to preclude such event, nor does he suggest that such events can be precluded. What he teaches is how to utilize processor resources when these long-latency events occur. Furthermore, Applicant respectfully disagrees with the Examiner's conclusion that it would consequently have been obvious to one of ordinary skill in the art at the time of the invention to modify the thread-switching system of Parady to include a hit/miss predictor as taught by McFarling, in order to switch thread before the load reaches the execution stage, thereby preventing a drop-off in throughput. As noted above, the only indication that Parady provides in his disclosure that a thread switch is required is an indication that a cache miss has occurred. Consequently, Applicant respectfully asserts that one skilled in the art at the time of the invention would not have been motivated to add the predictor of McFarling to the coarse-grained multithreaded processor of Parady because Parady provides no motivation or even an allusion that long-latency events can be precluded. Indeed, a long-latency event is required for Parady's invention to work. In addition, McFarling provides no motivation to one skilled in the art to incorporate his invention into a multithreaded processor. Applicant has searched McFarling and finds that he utterly fails to provide any impetus to applications other than the combination of two predictors in a single-threaded pipeline.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection of claim 1.

With respect to claims 2-5 and 14-16, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by Parady, McFarling, or Parady and McFarling in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 2-5 and that claims 14-16 be allowed following reconsideration..

Claim 6 is provided below for ease of reference.

6. A processor having multiple hardware steams supporting multiple data threads, the processor comprising:

a data cache, comprising a plurality of levels;

multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said data cache, said each of said multiple hit/miss predictors comprising:

a plurality of hit/miss predictors, each configured to forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss one or more of said levels; and

a fetch algorithm, coupled to said multiple hit/miss predictors, configured to select, on a cycle-by-cycle basis, the selected one of the multiple hardware streams from which to fetch the instructions, wherein said fetch algorithm selects the selected one of the multiple hardware streams based upon whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said one or more of said levels.

In a manner similar to claim 1, claim 6 recites, in combination with other elements, a fetch algorithm, coupled to multiple hit/miss predictors, configured to select, on a cycle-by-cycle basis, the selected one of the multiple hardware streams from which to fetch the instructions.

In the rejection of claim 6, the Examiner noted that Parady has taught a processor having multiple hardware streams supporting multiple data threads, comprising: a) a data cache; b) a fetch algorithm for selecting from which stream to fetch instructions. The Examiner noted that Parady has not taught a hit/miss predictor for predicting whether instructions

will hit or miss the cache wherein a prediction by the hit/miss predictor is used by the fetch algorithm in determining from which stream to fetch, but that McFarling has taught a hit/miss predictor that is used to predict, for load instructions, whether a cache hit or miss will occur. And, if a cache miss is predicted to occur, then instructions independent of the load are scheduled ahead of the load-dependent instructions. The Examiner states that a person of ordinary skill in the art would have recognized that this prediction scheme would be useful in a multiple thread environment because a thread is a sequence of instructions that is independent from other threads, as is known in the art. The Examiner continued that by implementing such a prediction scheme into the system of Parady, thread switches can occur sooner, thereby maximizing efficiency through execution of load-independent instructions. The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the thread-switching system of Parady to include a hit/miss predictor as taught by McFarling, in order to switch threads before the load reaches the execution stage, thereby preventing a drop-off in throughput.

For reasons substantially noted above in arguments presented in traversal of the Examiner's rejection of claim 1, Applicant asserts with respect to the rejection of claim 6 that Parady teaches a coarse-grained thread switching invention that must utilizes the indication of the occurrence of a long-latency event to trigger a thread switch. Applicant's invention deals with an entirely different application, that is, selection, on a cycle-by-cycle basis, of one of the multiple hardware streams from which to fetch instructions. And as noted above, Parady teaches away from the application area of the present invention by utilizing an indication that a cache miss has already occurred. Parady accepts that such events must occur after instructions are dispatched. In addition, Applicant has searched McFarling to find any teaching therein that would stimulate one skilled in the art to provide his predictor in the context of a fine-grained multithreaded processor. Applicant finds that McFarling is silent on any application of his invention beyond that of a single-threaded pipeline.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection of claim 6.

With respect to claims 7-10 and 17, these claims depend from claim 6 and add further limitations that are neither anticipated nor made obvious by Parady, McFarling, or Parady and McFarling in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 7-10 and that claim 17 be allowed following reconsideration.

Claim 11 is repeated below.

11. In a processor having multiple hardware streams supporting multiple data threads, and a data cache, a method for fetching instructions from a selected one of the multiple hardware streams to a pipeline, the method comprising:

for each of the multiple hardware streams, making a hit/miss prediction by a corresponding one of associated hit/miss predictors as to whether corresponding instructions for the each of the multiple hardware stream previously fetched will hit or miss the data cache; and

selecting, on a cycle-by-cycle basis, the selected one of the multiple hardware streams from which to fetch the instructions.

Like claims 1 and 6, claim 11 recites, in combination with other elements, selecting, on a cycle-by-cycle basis, the selected one of the multiple hardware streams from which to fetch the instructions.

In the rejection of claim 1, the Examiner noted that Parady has taught a processor having multiple hardware streams supporting multiple data threads, and a data cache, a method for fetching instructions from individual ones of multiple streams as instruction sources to a pipeline. The Examiner noted that Parady has not taught making a hit/miss prediction by a predictor as to whether instructions previously fetched will hit or miss the data cache and if the prediction is a miss, altering the source of the fetch, but that McFarling has taught a hit/miss predictor that is used to predict, for load instructions, whether a cache hit or miss will occur. And, if a cache miss is predicted to occur, then instructions independent of the load are scheduled ahead of the load-dependent instructions. The Examiner states that a person of ordinary skill in the art would have recognized that this prediction scheme would be useful in a multiple thread environment

because a thread is a sequence of instructions that is independent from other threads, as is known in the art. The Examiner continued that by implementing such a prediction scheme into the system of Parady, thread switches can occur sooner, thereby maximizing efficiency through execution of load-independent instructions. The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the thread-switching system of Parady to include a hit/miss predictor as taught by McFarling, in order to switch threads before the load reaches the execution stage, thereby preventing a drop-off in throughput.

Again, Applicant respectfully refers the Examiner to arguments presented above in traversal of the rejections of claims 1 and 6, and notes that Parady teaches an invention that is only provided for coarse-grained processors, that is, those processors that switch threads in response to the occurrence of a long-latency event. Applicant's invention, on the other hand, provides a method for fetching instructions from a selected one of multiple hardware streams to a pipeline, that operates on a cycle-by-cycle basis. As such, Parady teaches nothing that is relevant to the present invention for to utilize Parady's invention, a cache miss must occur when an instruction is executed. Applicant's invention provides a method for switching threads prior to when an instruction is dispatched. Likewise, McFarling fails to teach any application of cache miss prediction that other than that of a single-threaded pipeline.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection of claim 11.

With respect to claims 12-13 and 18-20, these claims depend from claim 11 and add further limitations that are neither anticipated nor made obvious by Parady, McFarling, or Parady and McFarling in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 12-13 and that claims 18-20 be allowed following reconsideration.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-20 are in condition for allowance. Reconsideration of the rejections and consideration of the new claims are requested, and allowance of all claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone at the direct dial number provided if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: **EO 002 582 768 US**
DATE OF DEPOSIT: **9/7/2004**
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop **PETITION**, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,
HUFFMAN PATENT GROUP, LLC

By _____
RICHARD K. HUFFMAN
Reg. No. 41,082
Tel.: (719) 575-9998

Date: 9/7/04

Attachments